GRAPH BASED MODELS TO DESCRIBE HIGH DIMENSIONAL TIME SERIES DATA

Benjamin Choi
ESE
WashU
St. Louis, MO 63112
bchoi22@wustl.edu

Argho Datta
ESE
WashU
St. Louis, MO 63112
argho.datta@wustl.edu

May 7, 2020

ABSTRACT

Much research has been conducted studying the movement of financial products as time series data. These products which move as a stochastic process are well suited to be studied using a parametric models such as ARIMA and are well-defined using such models. It is reasonable to see if two time series can be formulated as a mapping of one another. One way to do this would be to determine whether these two time series are *similar*. We would like to provide a novel definition of event synhcrony between two time series, and use financial data as a motivating example. To decide which financial products should be compared to one another, we use, as a prior, a dendogram. In addition to providing us a heuristic to choose which products to compare, this visual also gives researchers in industry a higher level representation of the relationship between products.

1 Objectives

1.1 High Level Overview

Financial products are well defined as stochastic processes. The most common equations used to describe these processes are the *Langevin Equations* and more specifically as the *Ornstein-Uhlenbeck Process* (OU). Such processes are known to have a mean reverting quality such that they revert to their mean function over time. The OU process is also known to be a continuous analog to an Autoregressive 1 (AR(1)) time series process. It is of interest to people in both industry and research to study whether two time series can be studied by defining a mapping between the two time series. To do this, one must define an event of Synchrony.

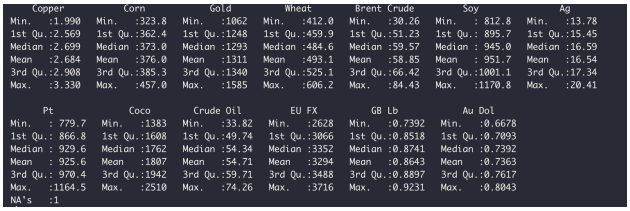
Our project proposes a novel definition for event synchrony as well as a method for detecting event synchrony between two time series.

We also provide researchers in industry a high level representation of time series data through a graph structure – namely a dendogram. These figures are generated by calculating pairwise distances between every combination of commodities.

1.2 Data Source

Our data was curated from Quandl, who provides an API for us to load the data directly into R, although API's exist for Python use as well. The data in question is commodity prices curated from the Chicago Mercantile Exchange and Chicago Board of Trade. Various products were included ranging from currencies, to metals, to agricultural products. The choice was not arbritary but because this was some of the only free data available using Quandl. Other data can be used should an individual have the appropriate Quandl Subscription.

We believe this high dimensional time series contains some underlying structure. Using a network representation of this data, we would like to extract some meaningful structure from the data. The raw data is summarized in Figure 1.



Copper	Corn	Gold	Wheat	Brent Crude	Soy	Ag
Min. :1.990	Min. :323.8	Min. :1062	Min. :412.0	Min. :30.26	Min. : 812.8	Min. :13.78
1st Qu.:2.569	1st Qu.:362.4	1st Qu.:1248	1st Qu.:459.9	1st Qu.:51.23	1st Qu.: 895.7	1st Qu.:15.45
Median :2.699	Median :373.0	Median :1293	Median :484.6	Median :59.57	Median : 945.0	Median :16.59
Mean :2.684	Mean :376.0	Mean :1311	Mean :493.1	Mean :58.85	Mean : 951.7	Mean :16.54
3rd Qu.:2.908	3rd Qu.:385.3	3rd Qu.:1340	3rd Qu.:525.1	3rd Qu.:66.42	3rd Qu.:1001.1	3rd Qu.:17.34
Max. :3.330	Max. :457.0	Max. :1585	Max. :606.2	Max. :84.43	Max. :1170.8	Max. :20.41

Pt	Coco	Crude Oil	EU FX	GB Lb	Au Dol
Min. : 779.7	Min. :1383	Min. :33.82	Min. :2628	Min. :0.7392	Min. :0.6678
1st Qu.: 866.8	1st Qu.:1608	1st Qu.:49.74	1st Qu.:3066	1st Qu.:0.8518	1st Qu.:0.7093
Median : 929.6	Median :1762	Median :54.34	Median :3352	Median :0.8741	Median :0.7392
Mean : 925.6	Mean :1807	Mean :54.71	Mean :3294	Mean :0.8643	Mean :0.7363
3rd Qu.: 970.4	3rd Qu.:1942	3rd Qu.:59.71	3rd Qu.:3488	3rd Qu.:0.8897	3rd Qu.:0.7617
Max. :1164.5	Max. :2510	Max. :74.26	Max. :3716	Max. :0.9231	Max. :0.8043
NA's :1					

Figure 1: Uncleaned Summary Statistics

2 Data Preprocessing

Because our data contains a mean and varaince that is a function of time, we make several conditions:

- Use a time series specific method for dealing with outliers
- Impose stationarity on the data

2.1 Cleaning Data

We elect to use the R function **tsclean()**. This uses STL decomposition to decompose a time series into a trend, seasonality and remainder component. Should the remainder component be extremely large, it will be flagged as an anomoly and replaced using linear interpolation. Values that are NaN in the original dataset are replaced using this method as well. An example of a cleaned, time series is shown as Figure 2.

Figure 2: Wheat Prices over our entire time course

2.2 Imposing Weak Sense Stationarity

Some of the distance methods used to generate our dendogram require that the data be stationary. Others require the looser condition of the data being time shift invariant. To impose stationarity for the data, we examine the time series X_t using the transformation $Y_t = (1 - D)log(X_t)$. We know that such a transformation gives us an almost exact approximation of the percent change in price (given that the change in price is sufficiently small). We provide both the transformed time series and the ACF to show the data is stationary. We observe the ACF decays exponentially to zero in lag. This is done for all commodities in our dataset.

2.3 Event in Time Series

In order to describe these two methods, we would first like to provide definitions for what an event is. We formalize our definition as the following.

For some time series $X = \{X_t | t \in T\}$, where T is the index set of times, we would like to define an *event* the time series over some interval during which the time series is wide-sense stationary.

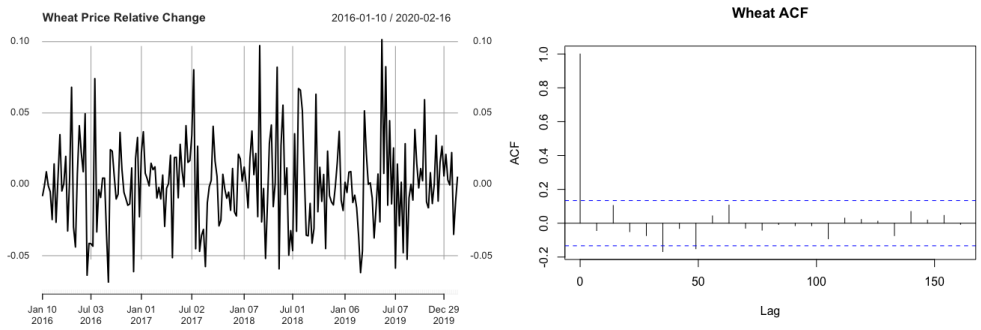


Figure 3: Plot of difference of log wheat prices and corresponding ACF

This condition will be tested for by studying both the ACF function as well as by using the Augmented Dickey-Fuller Test.

2.4 Motivation

Consider a stationary, zero mean time series Y_t and note that the random variables generated by $\{Y_t\}$, $t \in \mathrm{N}$ generates a Hilbert space $\mathcal{H}$. Such a time series can be interpreted, geometrically as a vector from the origin, as seen below.

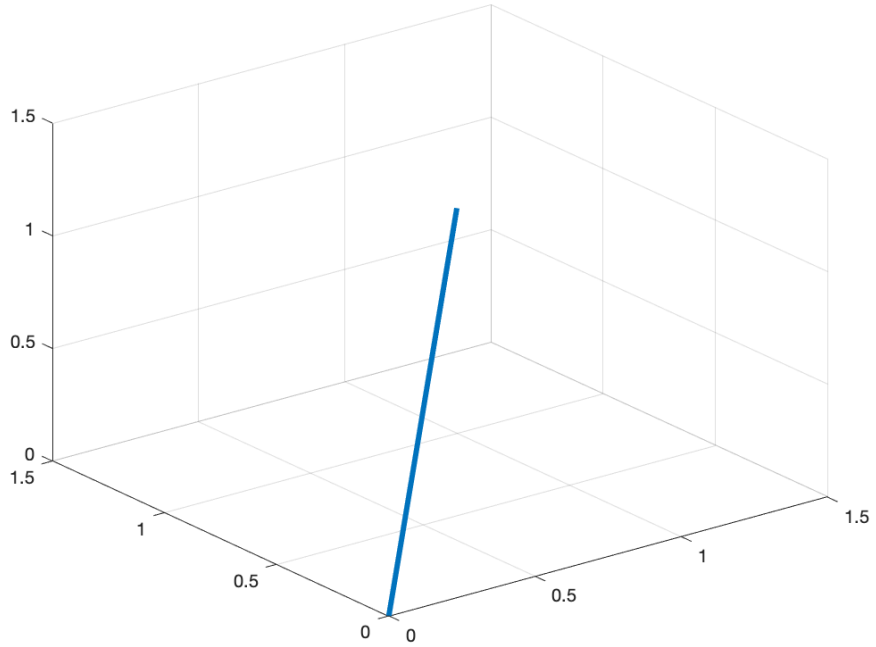


Figure 4: Time series in Hilbert space

In Hilbert Space, the Euclidean Distance is known to be a ball of some radius. Thus when we look to find two time series that are within the same neighborhood of one another, or are in the same ball. An example is given below with two time series X, Y.

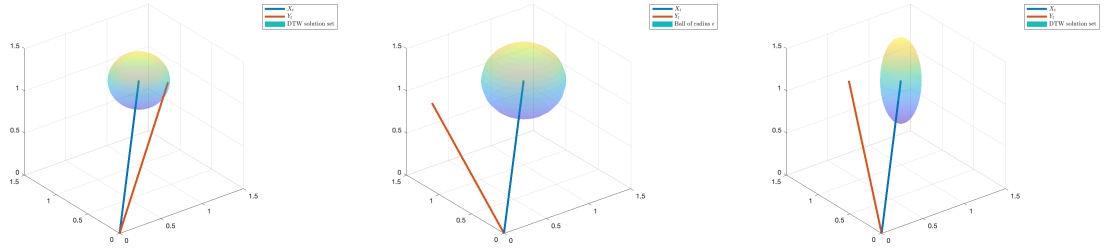


Figure 5: (From Left to Right) First we see a time series Y_t that is in the neighborhood of X_t wrt to Euclidean Distance and thus shows event synchrony. We next see a time series Y_t that is outside the region wrt Euclidean and does not show synchrony with X_t. Finally we see the solution set using the DTW distance metric.

2.5 Non-Parametric Method

We define event synchrony under this method to be as the following.

Let us define two time series $X = \{X_t | t \in T_x\}$ and $Y = \{Y_t | t \in T_y\}$, where T_x, T_y are the index sets for X, Y respectively. We note that the cardinality of these sets do not necesarrily have to be equal to one another.

We define an event of *synchrony* to be some time intervals where

$$\exists\, \epsilon \in R \;:\; d_(X_{t_1,t_1+\alpha}, Y_{t_2,t_2+\alpha}) < \epsilon \;\text{ where } t_1 \in T_x,\; t_2 \in T_y \tag{1}$$

2.6 Parametric Method

The parametric method for examining event synchrony imposes the following relationship. We define two time series $X = \{X_t | t \in T\}$, $Y = \{Y_t | t \in T\}$, $Z = \{Z_t | t \in T\}$, where T is the index set and Z is a white noise process. We define X to be a stationary ARIMA(p,d,q) process where

$$\phi(B)(1-B)^d X_t = \theta(B) Z_t \tag{2}$$

where B is the backshit operator, $\phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$, and $\theta(B) = 1 + \theta_1 B + \cdots + \theta_q \beta^q$. Also, through its, definition Z can be seen to be a stationary process. Our hypothesis is that Y is a stationary time series that follows an ARIMAX process of order (p,d,q,m) such that

$$\phi(B)(1-B)^d Y_t = \theta(B) Z_t + \psi(B) X_t \tag{3}$$

where $\psi(B) = \psi_1 B + \ldots \psi_m B^m$ and X is our exogenous variable.

We define $\hat{Y}$ to be the estimated ARIMA(p,d,q,m) process, generated using the following equation:

$$\hat{\phi}(B)(1-B)^d \hat{Y}_t = \hat{\theta}(B) Z_t + \hat{\psi}(B) X_t \tag{4}$$

where

$\hat{\phi}(B) = 1 - \hat{\phi}_1 B - \cdots - \hat{\phi}_p B^p$,

$\hat{\theta}(B) = 1 + \hat{\theta}_1 B + \cdots + \hat{\theta}_q \beta^q$, and

$\hat{\psi}(B) = \hat{\psi}_1 B + \ldots \hat{\psi}_m B^m$

are the estimated ARIMAX coeffecients. From our construction of $\hat{Y}_t$, it is easy to see that this process has a dependence on X. Furthermore, we observe the process $\hat{Y}$ is a stationary one by construction. We test our hypothesis of (3) by examining the synchrony that exists between processes $\hat{Y}$ and Y under the definitions we set in (1) and (**??**), such that

$$\exists\, \epsilon \in R \;:\; d(\hat{Y}_{t_1,t_1+\alpha}, Y_{t_1,t_1+\alpha}) < \epsilon \;\text{ where } t_1 \in T \tag{5}$$

3 Implementation

3.1 Change Point Detection

In order to determine whether the two time series are stationary, we examine their Auto Correlation Functions (ACF) in addition to their Augmented Dicky Fuller (ADF) statistic. Should the time series not be stationary over its entire time course, we propose using change point detection segment the time series into stationary segments.

3.2 Problem Framework

For the non-parametric method, we would like to identify synchrony by finding parameters t_1, t_2, α that satisfy the conditions set by (1) and (**??**). We thus look to solve the following minimization problem.
For two time series $X = \{X_t | t \in T_x\}$ and $Y = \{Y_t | t \in T_y\}$, where T_x, T_y are the index sets for X, Y respectively such that the two time series are stationary over these indexes. Consider a distance metric d.

$$\min_{t_1,t_2}\{d(X_{t_1,t_1+\alpha}, Y_{t_2,t_2+\alpha})\}, \;\; t_1 \in T_x, t_2 \in T_y \tag{6}$$

$$st. d(X_{t_1,t_1+\alpha}, Y_{t_2,t_2+\alpha}) < \epsilon$$

3.3 Algorithm

We solve the optimization following by first setting $t_1 = 1, t_2 = 1$. We then find a non-trivial α which will minimize our objective function. We then increase $t_2 = t_2 + 1$ and repeat the process. We then repeat the process with keeping $t_2 = 1$ and incriminating $t_1 = t_1 + 1$.

Algorithm 1: Syncrhony Detection

Result: t_1^*, t_2^*
Require: Two time series X, Y that are stationary over their entire respective time-courses, DTW tolerance ϵ, DTW Euc tolerance η, and window of time α;
Initialize: $t_1 = 0, t_2 = 0$;
while do
  while do
    $L = d_{DTW}(X_{t_1,t_1+\alpha}, Y_{t_2,t_2+\alpha})$;
    $M = L - d_{EUC}(X_{t_1,t_1+\alpha}, Y_{t_2,t_2+\alpha})$;
    if $L < \epsilon$ *and* $M < \eta$ **then**
      Record t_1, t_2
    end
    $t_2 = t_2 + 1$
  end
  $t_1 = t_1 + 1$
end

A visual is provided to explain how the algorithm works

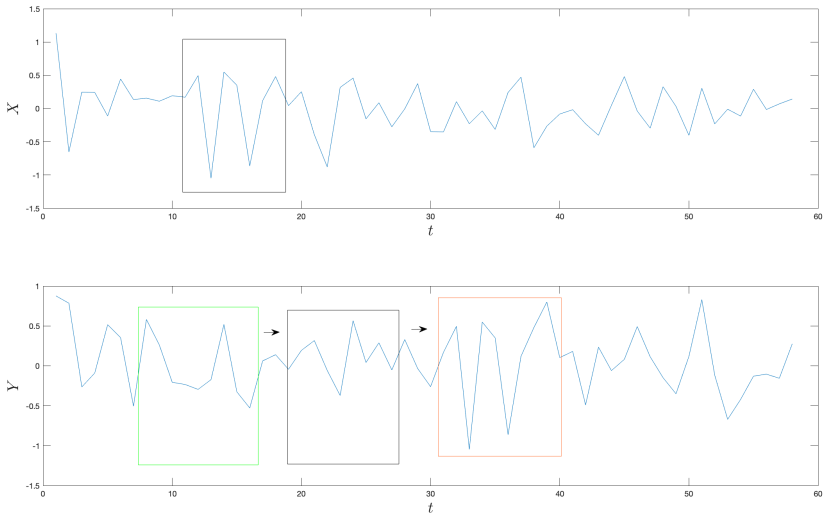


Figure 6: We consider event syncrhony over various windows of two time series

It is important to note that this method of obtaining time intervals is inherently a greedy and naive one. It is however, well suited for using Nonlinear Programming (NLP) or Genetic Algorithms (GA). Moreover, a problem arises in determining which pairs of products to compare, and one needs a method of determining the hyper-parameter α.

3.4 Genetic Algorithm

The GA algorithm is used to optimize solutions for a problem by selecting those values with the highest level of fitness. Through each iteration, It increases the chance for selecting these values by redefining the values associated with that fitness so that these values get put into the population for later use. Basically, this is how the GA algorithm redefines its search space without exhaustively going through all possible combinations

3.5 Using Dendogram as a Prior

We use dendograms to choose which products share strong similarities. Provides a visual for researchers to see pairwise and higher order relationships relationships. We find meaningful structure from the dendogram. Precious metals, agricultural products and petro products are grouped together. We next discuss how the dendograms are generated

4 Methods

4.1 Distance Methods

We study the following distance methods when generating the dendograms and find a heirachy of dissimilarity in products:

- Euclidian
- Dynamic Time Warping

A brief description for each method is provided below.

Euclidean Distance Measure

The Euclidean distance between two time series is equivalent to the square root of the sum of the squared length of the vertical lines between two time series that are sampled at the same frequency over the same time period.

Dynamic Time Warping

This method finds a mapping r such that the distance between coupled observations is minimized. More rigorously, this can be cast as the optimization problem

$$d_{DTW}(X_t, Y_T) = \min_{r \in M} \{ \sum_{i=1,\ldots,m} |X_{a_i} - Y_{b_i}| \} \tag{7}$$

where $X_T = [X_1, \ldots, X_n]$, $Y_T = [Y_1, \ldots, Y_n]$, M is the set of all possible sequences of pairs, and $r = ((X_{a_1}, Y_{b_1}), \ldots, (X_{a_m}, Y_{b_m}))$. The reference for this method is provided as [1] to the reader. It is, however, important to note that this dissimilarity metric is invariant to transforms of scaling and shifting to the data.

4.2 Clustering Method

As a control, we use Ward's Method, an agglomarative hierarchical clustering method to cluster the commodities. We elect to use the base method found in the R **hclust** package.

5 Analysis of Distance Metrics

5.1 Dissimilarity Heatmaps

We show below our preliminary results using our above methods using heatmaps:

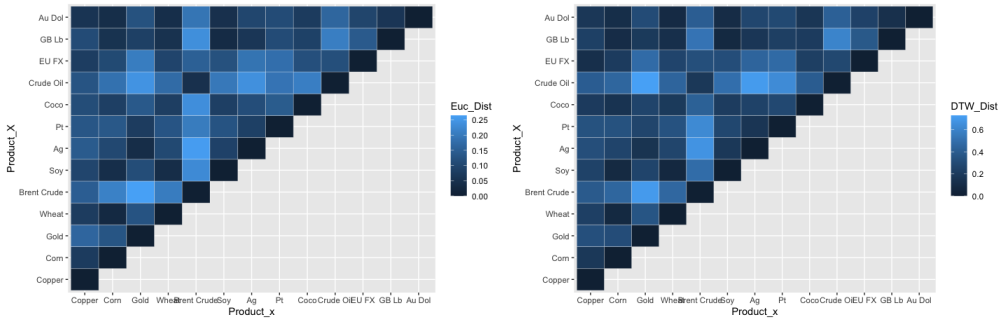


Figure 7: Euclidian and Dynamic Time Warping dissimilarity matrices

We notice that similarity between certain products are maintained despite suing different distance methods. Most notable is the high similarity between petro products {Brent Crude, Crude Oil}. This is better represented, however, using a dendogram.

5.2 Dendograms

Below, we present our preliminary dendograms: Here we observe that certain groupings are maintained through most of

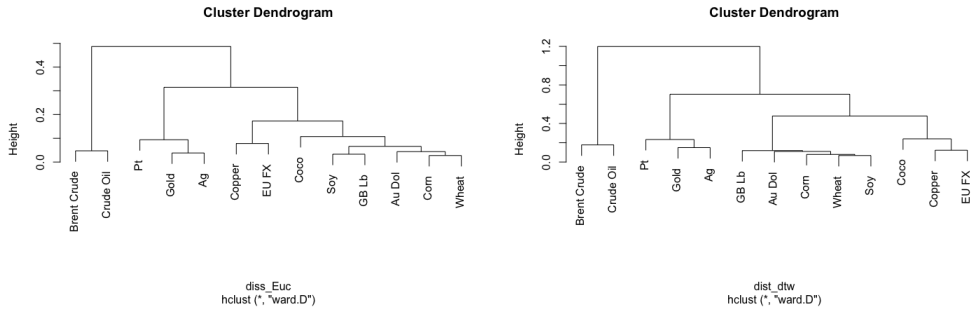


Figure 8: Dendograms generated using Euclidian and DTW distance metrics

the distance metrics. Even more interestingly, we notice that products that share similarities to *us* are grouped together. Indeed, petro products, precious metals, and agricultural are usually neighbors in the dendogram.

6 Results

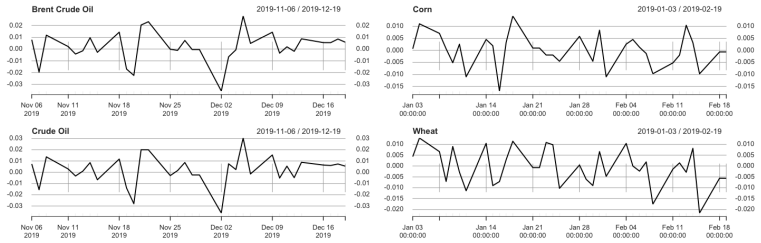


Figure 9: Event Synchrony between time series of Brent Crude Oil and Crude Oil, and between Corn and Wheat

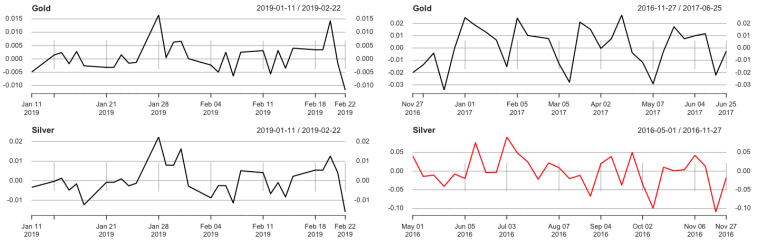


Figure 10: Event Synchrony between time series of Gold and Silver, while using the naive greedy algorithm on the left and the GA on the right

We observe certain interesting results. We observe that the movement of X and Y is very close from a qualitative look. Looking at overall distance between the two series, however, will provide more information. Oftentimes, the algorithm finds that, for two similar products, the interval of event synchrony occurs during the same time interval. Also, further work must be done to provide output of multiple populations from GA to see all values that satisfy ϵ condition.

7 Next Steps

We would like to be more rigorous in determining a distance metric is most suitable to our problem, ideally one that is invariant to change in dimension. Namely, Batista, Wang, Keogh provide a study of dissimilarity metrics by observing their invariance to transforms such as "local scaling (warping), uniform scaling, offset, amplitude scaling, phase, etc." [1] [2]

Also, we have discussed with Professor about applications to this algorithm, in bio medicine.

Ideally we would also like to study how the method can be used to study similarity between simulated time series, as described in (4). Much work is left to do, and we are excited to further this project.

References

[1] Pablo Montero and Jose A. Vilar. TSclust: An R Package for Time Series Clustering. In *Journal of Statistical Software*, November 2014, Volume 62, Issue 1.

[2] Batista GEAPA, Wang X, Keogh EJ (2011). "A Complexity-Invariant Distance Measure for Time Series." In*Proceedings of the Eleventh SIAM International Conference on Data Mining* 2011 SDM'11, pp. 699–710. SIAM, Mesa.